# Wireless Sensors Synchronization : an accurate and deterministic GPS-based algorithm

Vincent Le Cam, Arthur Bouche, David Pallier

Title: **Wireless Sensors Synchronization : an accurate and deterministic GPS-based algorithm**

Authors : Vincent Le Cam
Arthur Bouche
David Pallier

**ABSTRACT**

Last decade, the need for wireless sensors solutions as core-solutions of Structural Monitoring gained in interest. The cost of wireless devices compared to the cost of wiring important structures (bridges, energy-plants,...) is attractive. Most of recent development in WSN domain focused on energy (saving or harvesting), on wireless protocols, on embedded algorithms.

But it is a fact that, most of monitoring applications need samples to be time-stamped. According to the application, the wished time resolution could be up to one second for automation monitoring, one millisecond for vibration, one microsecond for acoustic monitoring, one nanosecond for electricity or light propagation...

The consequence for a Wireless network of electronic nodes is that, by nature, no common signal could physically provide a synchronization top. But, as each electronic device, a wireless sensor time-base uses a timer incremented by a quartz whose initial value is theoretical up to some p.p.m. and whose period drift on time because of age, temperature,...

Two kind of solutions could be regarded : a synchronization signal provided by the wireless protocol itself; an absolute synchronization from a referential source such as: GPS, Frankfurt clock, Galileo,... In the first way, it will be demonstrated the poor accuracy and the need of energy such a mechanism offers. In the second way, the article will details how a deterministic (Universal Time), accurate and resilient algorithm has been implemented.

The article also provides specific results of application on acoustic monitoring system and electricity propagation where the accuracy of a WSN has reached up to 10 nanosecond UT.

Consequence on energy consumption of this algorithm are given with a description of future works to improve the energy balance while keeping the device sober and synchronized.

IFSTTAR, COSYS, SII / Inria, I4S, Route Bouaye, Bouguenais, France

## INTRODUCTION

Whatever crystal quartz oscillator is initially chosen to design a wireless sensor, its accuracy will not be perfect and will lead to time-drift ( quartz at 1 ppm : one second every 11 days; standard quartz at 100 ppm : one second every 2.8 hours). Moreover this unstability increases with the device's age [1] and temperature [2].

Most of the time, time-synchronization of a wireless sensor network is either not implemented on the sensors level or based on a inter-node protocole. In the first case, data provided by wireless nodes are time-stamped when received by the supervisor (PC, data-logger, or the local bridge that foward data from the wireless LAN to the WAN). Thus, as an evidence, data are not time-samped accurately because of the network latency or the fact that data shipment from nodes is not made immediatly after data acquisition (bufferization).

In the second case, when wireless network synchronization is taken into account at node level, several techniques [3] [4] [5] [6] are proposed to share a mutual time reference across the network. Some of them became a standard like : NTP (Network Time Protocole), HRTS (Hierarchy Referencing Time Synchronization), TPSN (Timing-Sync Protocol for Sensor Nets), RBS (Reference Broadcast Synchronization),...But, as a matter of fact, those protocoles usually present the following defects :

- an increased power-consumption as the need for wireless beacons is important (in general)
- relative non deterministic efficiency due to network state, distances, topology... especially when implemented on multi-hop architectures (such as 802.15.4)
- best results that rarely exceed tens of microseconds

Thus, IFSTTAR has chosen another method that offers accuracy (up to nanoseconds) and determinism (no dependance on wireless network). This method resides in the use of an external source of synchronization, not depending on the wireless protocole : the PPS (Pulse Per Second) signal that output from each GPS receiver (other sources could have been regarded: Frankfurt clock, Galileo...).

## GPS-PPS SYNCHRONIZATION PINCIPLE

GPS-PPS synchronization principle is quite simple : as a standard, each GPS receiver delivers, each second, a binary signal named PPS (Pulse Per Second). PPS is an absolute time source that is very stable in time (nearly constant period) and nearly absolute as it is generated from atomic-clock inside GPS satellites. Most of the manufacturers (Trimble, UBlox, Leica...) offer GPS receivers with PPS accuracy ranging from 10 to 100 nanoseconds. Thus, a first implementation could be done as illustrated in figure 1:

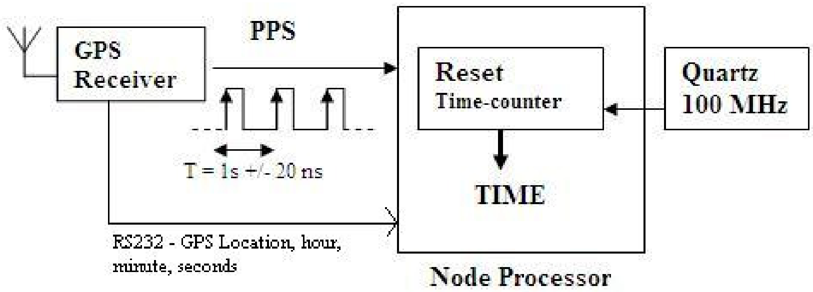


Figure 1. Typical PPS implementation in a Wireless node.

But, when implemented on smart sensors using operating systems like Linux or advanced software stacks, there is a miss of accuracy due to Interruption (IT) latency typically ranging from 1 to 100 mircoseconds for traditional microprocessors.

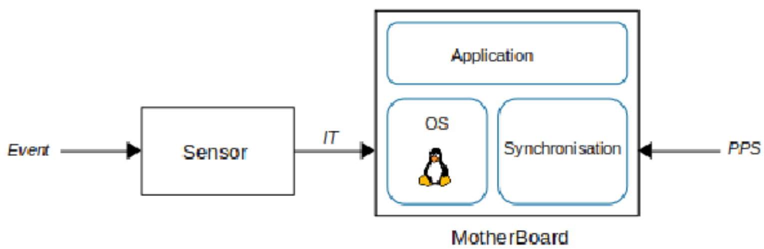


Figure 2. Event detection latency on smart nodes.

Based on these statements, presented work in this article aims to improve the accuracy when time-stamping unpredictable events. This work has been implemented using PEGASE 2 Ifsttar wireless platform.

## IMPLEMENTATION ON PEGASE 2 WIRELESS PLATFORM

### PEGASE 2 platform description

PEGASE is a generic Wireless Sensor Platform conceived by IFSTTAR. The project started in 2008. PEGASE concept is based on a generic vision of its hardware and software abilities. Hardware genericity is provided by a principle of mother/daughter pluggable boards. PEGASE mother board integrates common functions of typical wireless systems: computation, energy, multiple I/Os and wireless communications. Each pluggable daughter board adds specific functions to the mother board, such as: 8-analog/digital channels, 3G GSM extension, accelerometers... Software genericity is obtained through a Linux Operating System added to a Software Development Kit (SDK) given in C++ open-source language. A first generation of PEGASE 1 has been sold by a third-party company of IFSTTAR in thousands of units since 2008. It is used in many SHM applications (acoustic monitoring of bridge cables, strain gauges or vibration monitoring, etc.) As electronic is a domain subjected to fast evolutions, a new PEGASE generation is about to be set up. PEGASE 2 is not only a more efficient electronic device, but it is also linked to a cloud supervision software that allows to operate various sensors.

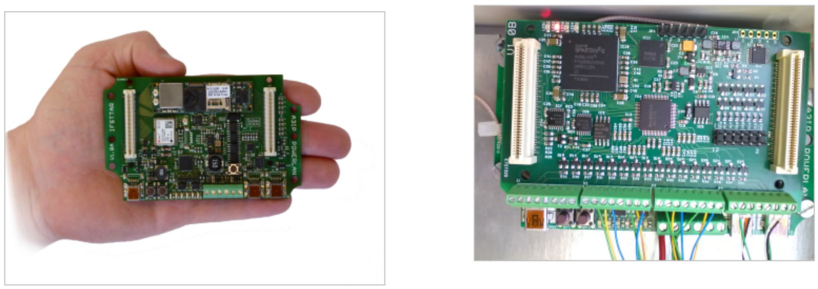

Figure 3. PEGASE2 mother and 8 channel duaghter boards.

PEGASE-2 eight-channels daughter-board proposes 8 differential analog inputs sampling on 24 bits up to 50 kHz (accelerometers, strain gauges, temperature, ...) and 8 digital inputs. Based on a FPGA synonym of determinism in signal transitions, this platform has been chosen for the synchronization improvements.

## Works, improvements and results on PEGASE 2 platform

### DESIGN IMPROVEMENTS

The eight-channels daughter-board is designed around an FPGA that counts time using a 2 p.p.m crystal quartz oscillator at 240MHz. FPGA timeCounter is reset on the rising edge of each PPS signal from the GPS receiver. When an event occurs, the corresponding step of the timeCounter is **immediately** stored in the internal FPGA memory and the FPGA raises an interruption (IT) for the mother board. The mother-board can then asynchronously get the event and associated timestamps and recompute the date of the event using :

$$T_i = timeCounter_i/240000000 \tag{1}$$

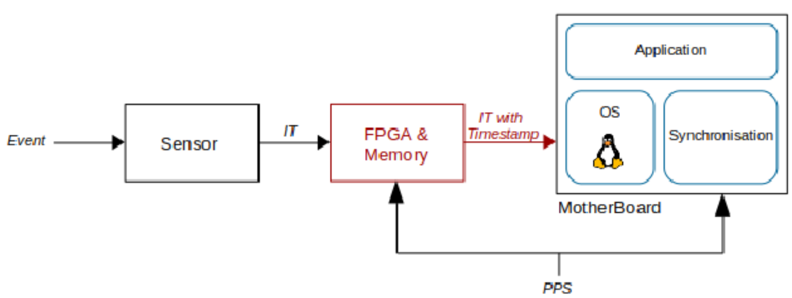


Figure 4. FPGA based design with shared PPS.

This method avoid delay in timestamping due to OS overhead and converts the non-deterministic time between the actual event and the timestamping into a deterministic delay being implemented on an FPGA.

## USING THE QUARTZ REAL FREQUENCY

As seen in introduction, crystal quartz oscillator frequency changes as it ages [1] and can be disturbed by external factors such as temperature [2], accelerations [7] or magnetic fields [8]. Thus, even if the system is synchronized on PPS edges, FPGA counter can vary. IFSTTAR focused on temperature since it is the main disturbance in our applications. The real frequency f (compared to its theoritical frequency f0) of a quartz crystal oscillator at x ppm relative to the real temperature T of this oscillator is given by the following formula:

$$f = f_0.(1 - x.ppm(T - T_0)^2) \tag{2}$$

However, in most of our use cases temperature does not vary very significantly during a second. Thus, instead of controlling the quartz according to the ambiant temperature (that requires an accurate temprature sensor at quartz level) each second, the linux OS gets the FPGA frequency during the previous second. On IT, when the OS ask for events and associated timestamps to the FPGA, it will re-compute the date with the right frequency fi:

$$T_i = timeCounter_i.f_i \tag{3}$$

## USING GPS IN FIXED MODE

Each GPS receiver compute its position and the time by solving permanently the equation below:

$$P = \sqrt{(X_s - X)^2(Y_s - Y)^2(Z_s - Z)^2} + I + T + c.\Delta_\tau + E \tag{4}$$

This equation depends on four variables : three antenna's coordinates X, Y, Z and the synchronization error $\Delta_\tau$. The terms I and T are negligible because they are systematic errors that don't evolve with short time intervals. At least four equations are necessary for the GPS receiver to resolve the equation and find X, Y, Z and $\Delta_\tau$. When the GPS receiver is not in fixed mode, the solution of each variable is not accurate. However, in our case, sensor needs a highly accurate and resilient synchronization ($\Delta_\tau$).

To improve the time accuray, variables X, Y and Z can be set by means of a transition to Survey-in mode. Entering Survey-in mode is done by setting a minimum observation period and a maximum position uncertainty. Both constraints must be observed for the transition in "Fixed" mode. The following exemple defines typical constraints : 1 m and 24 hours. In this mode, the NEO 6T GPS of PEGASE 2 is self-located with accuracy. Once the GSP module is in "fixed" mode, the X Y and Z positions of the antenna are known very accuractely, the 4-unkown equations then turn into a one-unknown equation (delta). The PPS becomes therefore more accurate.

UTILISATION OF QERR

The rising edge of the PPS signal generated by the GPS receiver is synchronized by its internal clock. However, the PPS signal computed by the GPS receiver is not perfectly synchronized on atomic clocks of GPS constellation. Thus, some GPS receivers compute and provide the time difference between the PPS signal generated and the real PPS signal. This data (qErr) is in picosecond; qErr is provided 1 second after its computation. The figure below illustrates the evolution of the qErr during five minutes. The results are the same for longer periods. The variable qErr is typically between +10ns and -10ns and is centered on 100ps.

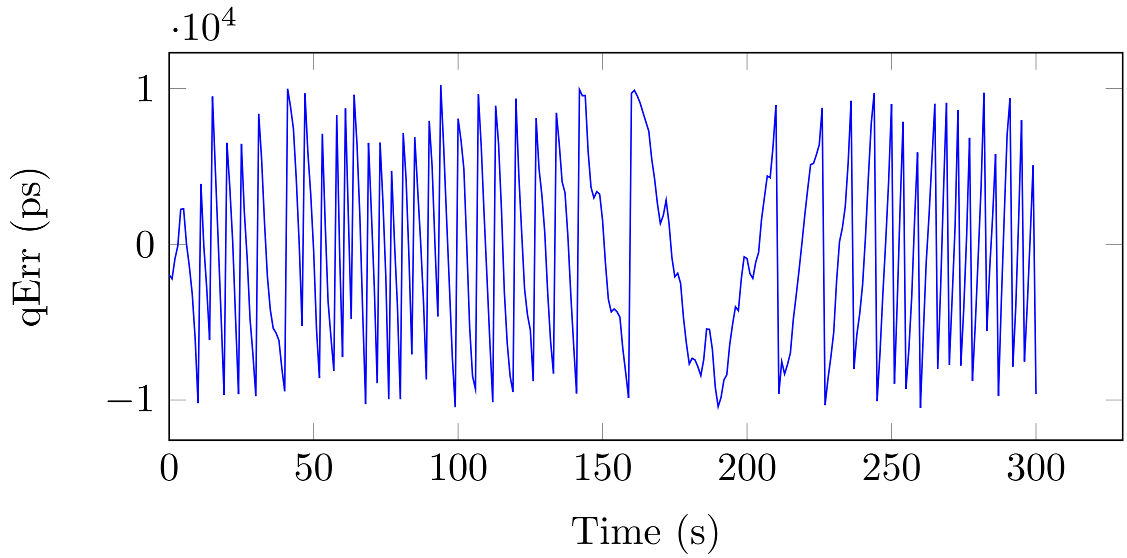


Figure 5. Evolution of qErr during 5 minutes.

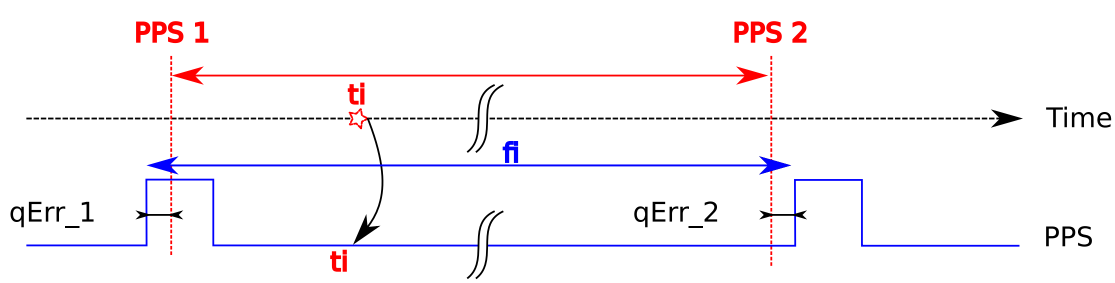


Figure 6. Exemple of date's recalculatting.

Figure 6 shows the different data available for the Linux OS when an event happen. The frequency doesn't match with the counter step in one second but with counter step less qErr1, more qErr2. Furthermore, the beginning of the second is un-synchronized of qErr1 in comparison to the theoretical PPS. IFSTTAR elaborated this formula to correct the time-stamp:

$$T_{ins} = 10^{-3}.qErr1_{ps} + t_i.f_i.(10^9.1_s - 10^{-3}.qErr1_{ps} + 10^{-3}.qErr2_{ps}) \quad (5)$$

The figure 7 shows the experiment that was set-up to observe the disparity of date's correction between two PEGASE2. A same edge event is injected on two boards and a comparison is made between the two dates. Figure 8 gives an example of date correction. The FPGA synchronized by PPS A counts a second longer than the FPGA synchronized by PPS B.

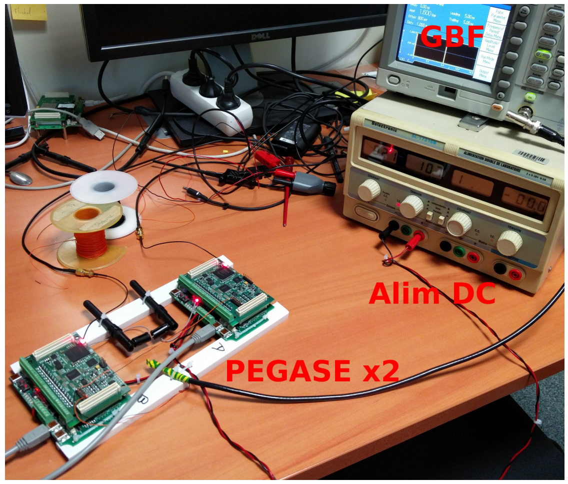


Figure 7. Test set-up.

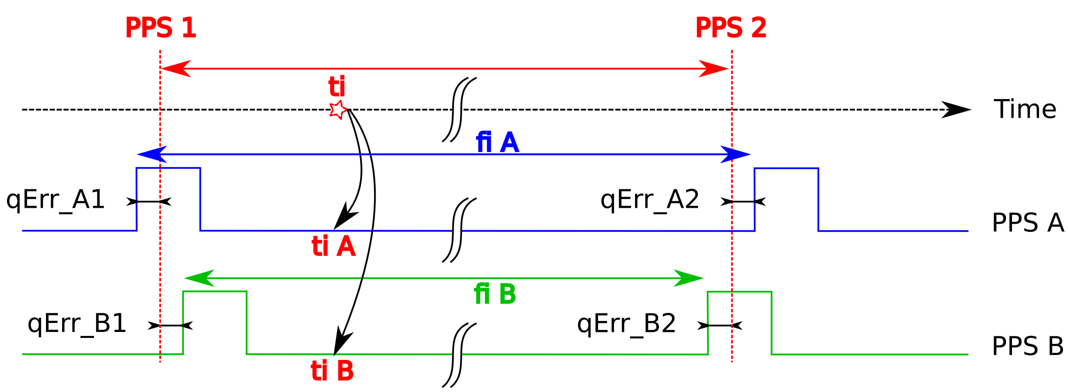


Figure 8. Illustration of date correction.

The results of the interval between tiA and tiB are given on figure 9, with and without qErr effect. Using qErr for computation of the date strongly increases the accuracy.

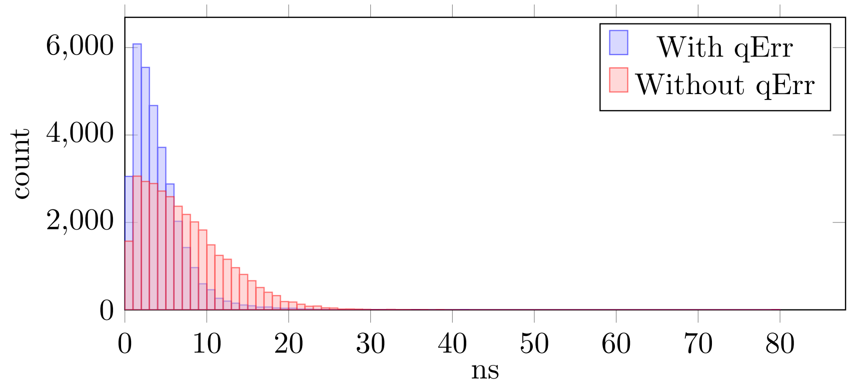


Figure 9. Interval in nanoseconds between tiA and tiB with and without qErr.

## CONCLUSION

This article illustrates how a GPS solution as source of time-synchronization for wireless sensors nodes could be relevant compared to traditionnal un-accuracte or un-deterministic solutions that use wireless inter-nodes protocols. IFSTTAR works consisted in using the PPS signal to have a stable time reference across the network but also to take into account, incrementally: temperature influence, fixed-mode of GPS receiver, qErr correction coefficients... These software and hardware integrations led IFSTTAR to a generic principle (e.g. not only valid in PEGASE2) where a same event can be time-stamped with an accuracy of around 6 nanoseconds by different sensors. Due to GPS technology, these results are nearly deterministic and stable in time whatever the wireless protocol chosen and the number and localization of the nodes. A first application aiming at detecting and localizing lightning on power-lines by time-stamping current pulses uses this concept since the pulses have to be time-stamped with an accuracy of less than 50 nanoseconds.

## PERSPECTIVES

While this paper has demonstrated the potential of GPS-PPS based deterministic timestamping, many opportunities for extending the scope and reliability of synchronization in wireless sensors network remain:

- The GPS-PPS based deterministic synchronization works for outdoor motionless systems but is it possible to achieve the same results with systems subjected to loss of GPS signal, fast temperature changes, accelerations, vibrations etc ? How to assess the reliability of the synchronization in systems subjected to such disturbences ?
- The use of a GPS is very energy expensive so another approach would be to drive the on/off cycle of the GPS based on the synchronization accuracy needed for the application (ns, us, ms) and the external disturbences in order to minimize power consumption.

The SENTAUR thesis (Sensor Enhancement To Augmented Usage and Reliability) supported by IFSTTAR and IETR will address these challenges.

## REFERENCES


1. Wang, S.-y., B. Neubig, K. Sato, T. Hosoda, E. Seydel, J.-h. Wu, T.-f. Ma, and J. Wang. 2015. "Aging models and parameters of quartz crystal resonators and oscillators," in *Piezoelectricity, Acoustic Waves, and Device Applications (SPAWDA), 2015 Symposium on*, IEEE, pp. 382–385.
2. Zhou, W., H. Zhou, Z. Xuan, and W. Zhang. 2005. "Comparison among precision temperature compensated crystal oscillators," in *Frequency Control Symposium and Exposition, 2005. Proceedings of the 2005 IEEE International*, IEEE, pp. 5–pp.
3. Le Cam, V. 2006. "Synchronization of wireless sensors: review of methodologies. experience feedback of the very precise GPS solution," in *Third European Workshop on Structural Health Monitoring*.
4. Elson, J. and D. Estrin. 2001. "Time synchronization for wireless sensor networks." in *IPDPS*, vol. 1, p. 186.
5. Djenouri, D. and M. Bagaa. 2016. "Synchronization protocols and implementation issues in wireless sensor networks: A review," *IEEE Systems Journal*, 10(2):617–627.
6. Sarvghadi, M. A. and T.-C. Wan. 2014. "Overview of time synchronization protocols in wireless sensor networks," in *Electronic Design (ICED), 2014 2nd International Conference on*, IEEE, pp. 204–209.
7. Filler, R. L. 1988. "The acceleration sensitivity of quartz crystal oscillators: a review," *IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control*, 35(3):297–305.
8. Brendel, R. 1996. "Influence of a magnetic field on quartz crystal resonators," *IEEE transactions on ultrasonics, ferroelectrics, and frequency control*, 43(5):818–831.